



SEGUNDA AVALIAÇÃO DE LINGUAGENS E COMPILADORES (PCS-2056)

Seja conciso e rigoroso na terminologia técnica.

A linguagem LazyComb é minimalista, funcional, com coletor de lixo (*garbage-collector*), com transparência referencial e com um sistema de E/S simples baseado em *stream*. A linguagem captura a essência da programação funcional e seus programas são definidos por meio do uso de **combinadores**. Um combinador executa uma ação que pode ser implementada através de uma abstração lambda.

Uma abstração lambda é uma especificação de função anônima com parâmetros como, por exemplo, a definição da função identidade, (*lambda* (*x*) *x*), assim a aplicação de um valor à identidade retornará o próprio valor: (*lambda* (*x*) *x*) 3 => 3. O combinador *I* tem exatamente essa função: *I* 3 => 3. Entretanto, não há valores numéricos, eles devem ser todos representados como funções, assim o numeral 0 é representado por (*lambda* (*f x*) (*f x*)), logo 1 é representado por (*lambda* (*f x*) (*f (f x)*)). Desta maneira qualquer número natural é representado pela composição funcional. Usando combinadores o número 256 fica **SII(SII(S(S(KS)K)I))**.

A sintaxe de LazyComb é:

Sintaxe	Semântica
Program ::= CCEExpr	CCEExpr
CCEExpr ::= CCEExpr Expr epsilon	(CCEExpr Expr) (lambda (x) x)
Expr ::= "i" Expr'	(lambda (x) x) Expr'
IotaExpr ::= "i" como abaixo Expr'	(lambda (x) (x S K)) ; com S e K definidos Expr'
Expr' ::= "I" "K" "k" "S" "s" NonemptyJotExpr "`" Expr1 Expr2 "*" IotaExpr1 IotaExpr2 "(" CCEExpr ")"	(lambda (x) x) (lambda (x y) x) (lambda (x y z) ((x z) (y z))) NonemptyJotExpr (Expr1 Expr2) (IotaExpr1 IotaExpr2) CCEExpr
NonemptyJotExpr ::= JotExpr "0" JotExpr "1"	(JotExpr S K) (lambda (x y) (JotExpr (x y)))
JotExpr ::= NonemptyJotExpr epsilon	NonemptyJotExpr (lambda (x) x)

Considerando que a execução de um programa em LazyComb é baseada na aplicação de combinadores a expressões, ou seja, cada programa será executado a partir de uma lista prefixada de operadores que terminará¹, pede-se (Deverá ser entregue no dia 1/12/2009 às 10h necessariamente, sem ele perde-se 2,5 pontos.):

1. (2,5 pontos) Construa um reconhecedor determinístico, baseado no autômato de pilha estruturado, que aceite como entrada válida um programa em LazyComb.

¹ Na verdade o programa também pode entrar em um processo infinito de busca, mas isso não é relevante neste momento